

Multiplatformní vývoj mobilních aplikací

Cross-platform Mobile Application Development

Zadání bakalářské práce

Student:

Michal Seidler

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Multiplatformní vývoj mobilních aplikací
Cross-platform Mobile Application Development

Zásady pro vypracování:

Cílem práce je seznámit se s možnostmi vývoje aplikací pro mobilní zařízení s ohledem na přenositelnost zdrojových kódů pro různé platformy, detailně se zaměřit na framework PhoneGap a vytvořit mobilní verzi administračního systému webové aplikace pro komunikaci s klienty. Aplikace bude obsahovat práci s kontakty z firemní databáze (vkládání kontaktů, organizování do kontakt listů), rozesílání emailové pošty na dané kontakty včetně analýzy (počet doručení, otevření, zobrazení příloh), přímé telefonování a monitorování délky hovoru. Bude také umožněno komunikovat prostřednictvím mobilního telefonu se zákazníky na webových stránkách pomocí online chatu. Aplikace bude zaměřena na platformy WP7, Android případně iOS.

Seznam doporučené odborné literatury:

- [1] WARGO, John M. PhoneGap essentials: building cross-platform mobile apps. 2012. vyd. Upper Saddle River, NJ: Addison-Wesley, c2012, p. cm. ISBN 03-218-1429-0.
- [2] LUNNY, Andrew. Phonegap beginner's guide: build cross-platform mobile applications with the PhoneGap open source development framework. Birmingham, UK: Packt Publishing Limited. ISBN 978-184-9515-368.
- [3] PILGRIM, Mark. Dive into HTML5, <http://diveintohtml5.info/>.
- [4] FLING, Brian. Mobile design and development. 1st ed. Sebastopol, Calif.: O'Reilly, c2009, xix, 309 p. ISBN 05-961-5544-1.
- [5] NEIL, Theresa. Mobile design pattern gallery. 1st ed. Sebastopol, CA: O'Reilly, c2012, xii, 261 p. ISBN 14-493-1432-5.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Holíš**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2013

A handwritten signature in black ink, appearing to be 'G. J. ...', written over a dotted line.

Rád bych na tomto místě poděkoval všem, kteří mi pomohli a podpořili mě při vypracování bakalářské práce, jmenovitě Michalu Holišovi za profesionální přístup při vedení této práce a cenné rady, Lukáši Richtárovi a týmu mSolution.cz za zázemí a podporu a v neposlední řadě mým klientům za jejich shovívavost a porozumění.

Abstrakt

Cílem práce je seznámit se s možnostmi vývoje aplikací pro mobilní zařízení s ohledem na přenositelnost zdrojových kódů pro různé platformy, detailně se zaměřit na framework PhoneGap a vytvořit mobilní verzi administračního systému webové aplikace pro komunikaci s klienty. Aplikace bude obsahovat práci s kontakty z firemní databáze (vkládání kontaktů, organizování do kontakt listů), rozesílání emailové pošty na dané kontakty včetně analýzy (počet doručení, otevření, zobrazení příloh), přímé telefonování a monitorování délky hovoru. Bude také umožněno komunikovat prostřednictvím mobilního telefonu se zákazníky na webových stránkách pomocí online chatu. Aplikace bude zaměřena na platformy WP7, Android případně iOS.

Klíčová slova: multiplatformní vývoj, mobilní aplikace, PhoneGap, HTML5, JavaScript, CSS

Abstract

The aim of the thesis is to introduce options of mobile phones application development regarding portability of source codes for various platforms. Attention is given to PhoneGap framework and to a development of a mobile admin system of real web application, which is used to communicate with clients. Applications will include working with contacts from the corporate database (insert contacts, organizing into contact sheets), sending e-mail to contacts including analysis (number of delivery, open rate, view attachments), direct phone calls and monitoring call duration. It will also be able to communicate via mobile phone to customers on the website online chat. Application will be focused on WP7, Android and iOS platforms.

Keywords: cross-platform development, mobile applications, PhoneGap, HTML5, JavaScript, CSS

Seznam použitých zkratk a symbolů

OS	–	Operační systém
CRM	–	Customer relationship management
CDN	–	Content delivery network
API	–	Application Programming Interface
AJAX	–	Asynchronous JavaScript and XML
JSON	–	JavaScript Object Notation
JSONP	–	JavaScript Object Notation with Padding

Obsah

1	Úvod	5
1.1	Struktura práce	5
2	Vývoj mobilních aplikací	7
3	Nejběžnější mobilní operační systémy	8
3.1	Android	8
3.2	iOS	8
3.3	Windows Phone	9
3.4	BlackBerry	9
3.5	Bada	10
3.6	Symbian	10
3.7	WebOS	10
4	Možnosti univerzálního vývoje mobilních aplikací	11
4.1	Webové aplikace	11
4.2	Multiplatformní frameworky a nástroje	11
5	Framework PhoneGap	13
6	Vývoj multiplatformní aplikace ve frameworku PhoneGap	16
6.1	Popis webové aplikace	16
6.2	Komunikace mezi webovou aplikací a mobilní aplikací	17
6.3	Popis mobilní aplikace	20
6.4	Použité technologie a frameworky	20
6.5	Ukázková mobilní aplikace	21
7	Kompilace mobilní aplikace pro různé operační systémy	26
7.1	Kompilace aplikace pro operační systém Android	26
7.2	Kompilace pro operační systém Windows Phone	29
7.3	Problémy a jejich řešení	29
8	Závěr	30
9	Reference	31
	Přílohy	31
A	Uživatelské rozhraní aplikace	32
B	Příloha na CD/DVD	34

Seznam obrázků

1	Vývoj internetového provozu z mobilních zařízení[8]	6
2	Architektura frameworku PhoneGap [9]	13
3	Architektura služeb webové aplikace	17
4	Sekvenční diagram autentizace	19
5	Hello World! PhoneGap aplikace	28
6	Uživatelské rozhraní	32
7	Ukázka kreslení na Canvas	33

Seznam tabulek

1	Přehled PhoneGap API [10]	14
2	Přehled podporovaných funkcí v operačních systémech [10]	15
3	Přehled výkonu jednotlivých aplikací	25

Seznam výpisů zdrojového kódu

1	Implementace JSONp v serverové části	18
2	Jednoduché ověření fungování JSONp	18
3	Použití frameworku Angular JS pro JSONp	18
4	Nešifrovaný a šifrovaný požadavek	20
5	Ukázka implementace uživatelského rozhraní	22
6	Ukázka implementace aplikační logiky	22
7	Kostra PhoneGap aplikace	26
8	Třída hlavní aktivity Android aplikace	27
9	AndroidManifest.xml	27
10	Použití knihovny iScroll	29

1 Úvod

Mobilní zařízení jako chytré telefony a tablety přebírají značnou část internetového provozu. Tento trend můžeme sledovat především na statistikách meziročního růstu počtu vyhledávání na internetu, sledování videí a čtení emailu právě z těchto mobilních zařízení. Tyto statistické údaje poskytují zejména velké společnosti zabývající se internetovým provozem a technologiemi. Na grafu společnosti KPCB (viz obr. 1) můžeme vidět růst mobilního provozu v meziročním srovnání.

Růst internetového provozu z mobilních zařízení vychází především z jednoduchého, rychlého a nyní již globálně dostupného mobilního internetu. Díky možnosti sdílet data mezi vlastními i cizími zařízeními pomocí tzv. Cloudových¹ řešení se nemusíme omezovat počtem zařízení a máme svá data pohodlně kdykoliv k dispozici. Uživatelé osobních počítačů, chytrých telefonů a tabletů si možnost neustále dostupnosti a sdílení dat rychle osvojili a začali vyžadovat přístup k jejich oblíbeným aplikacím a službám také prostřednictvím jejich mobilních zařízení.

V bakalářské práci se věnuji problematice multiplatformního vývoje mobilních aplikací. Při vývoji multiplatformních mobilních aplikací je potřeba řešit nejen samotný vývoj, ale také navržení použití vhodných technologií s ohledem na znovu použitelnost zdrojových kódů pro migraci na další platformy. Pokud zvolíme správné technologie a nástroje můžeme díky tomu optimalizovat časové, ale také finanční náklady nutné pro rozšíření naší aplikace na další platformy.

Cílem mé práce je seznámit se s dostupnými technologiemi, porozumět jejich výhodám a navrhnout vhodnou kombinaci daných technologií pro maximální optimalizaci vývoje mobilní aplikace z časového a finančního hlediska. V praktické části bakalářské práce se budu zabývat migrací již existující webové aplikace na mobilní platformy. Součástí této migrace bude vytvoření komunikační API a propojení již běžících služeb² s touto API. Na závěr praktické části použiji stejné technologie zvolené pro migraci webové aplikace a pokusím se je využít při vytvoření hry Snake a také vykreslení fontány o 2000 částicích na Canvas. Na této ukázce ověříme, zda zvolené technologie mohou být použity pro různé typy aplikací, nebo je nutné vybírat technologie na základě toho, jakou aplikaci chceme vyvíjet.

1.1 Struktura práce

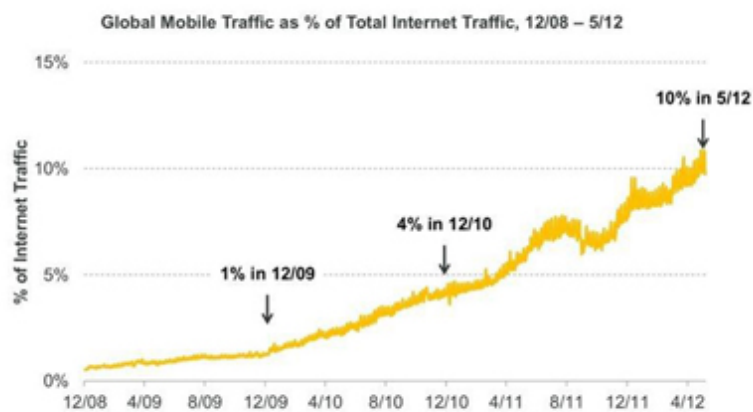
Práce se dělí na šest kapitol. Kapitola první popisuje problém vývoje mobilních aplikací pro různé mobilní platformy. V druhé kapitole představuji nejběžnější mobilní platformy, jejich historii a nativní programovací jazyky a vývojové nástroje. Třetí kapitola se zabývá možnostmi vývoje multiplatformních aplikací a jejich výhodami a nevýhodami. V této kapitole také představím jednotlivé frameworky a nástroje pro multiplatformní vývoj. Čtvrtá kapitola je zaměřena na popis frameworku PhoneGap. V páté kapitole popisují vývoj ukázkové mobilní aplikace ve frameworku PhoneGap, konkrétní příklady a řešení

¹Jedná se o poskytování služeb, programů či dat uložených na serverech na internetu s možností přístupu přes webový prohlížeč či klienta dané aplikace.

²emailové fronta, jabber server, analytická služba

problémů. V šesté a zároveň poslední kapitole uvádím příklady kompilace aplikace ve frameworku PhoneGap pro různé platformy.

Global Mobile Traffic Growing Rapidly to 10% of Internet Traffic



KPCB

Source: StatCounter Global Stats. 14

Obrázek 1: Vývoj internetového provozu z mobilních zařízení[8]

2 Vývoj mobilních aplikací

Od počátku éry mobilních platforem, kdy v roce 1996 vznikl první operační systém Palm OS vniklo do dnes přibližně 21 dalších operačních systémů pro mobilní zařízení. Řada těchto operačních systémů je již mrtvá a některé další postupně vymírají ovšem i tak je zde stále okolo 15 aktivních operačních systému, které jsou stále podporovány a vyvíjeny.

Většina těchto mobilních operačních systémů je založená na podobných často dokonce stejných technologiích, ale ve způsobu vývoje mobilních aplikací se liší. Každý z mobilních operačních systémů podporuje jiné programovací jazyky a disponuje rozdílnou API pro přístup k funkcím mobilního přístroje. Pokud se rozhodneme při vývoji mobilních aplikací zaměřit na dnes nejrozšířenější mobilní operační systémy pravděpodobně to budou Android, iOS, Windows Phone, BlackBerry, Bada, Symbian a WebOS. Pokud chceme vyvíjet nativní aplikaci pro všechny tyto mobilní operační systémy máme dvě základní možnosti. První z nich je naučit se pracovat se všemi platformami, jejich vývojovými nástroji a programovacími jazyky, nebo hledat alternativní řešení, které nám vývoj usnadní.

Mezi alternativní řešení jistě patří multiplatformní vývojové nástroje. Jedná se o nástroje a technologie díky kterým je možné vyvíjet aplikace za použití stejných zdrojových kódů. Multiplatformní nástroje fungují obvykle na dvou základních principech. Prvním z nich je takzvaný wrapper, který obalí zdrojový kód aplikace napsaný v univerzálním programovacím jazyce do nativní aplikace a zprostředkovává komunikaci mezi aplikací a mobilním přístrojem. Dalším principem je použití vlastního programovacího jazyka. Takto napsaná aplikace je poté kompilována na nativní aplikaci v nativním jazyce mobilní platformy.

3 Nejběžnější mobilní operační systémy

Mezi nejběžnější mobilní operační systémy patří Android, iOS, Windows Phone, BlackBerry, Bada, Symbian a WebOS. Tyto operační systémy se liší jak uživatelským rozhraním tak především funkcionalitou a provedením. I když mobilní operační systémy často bývají založeny na Linuxovém jádře většina z nich používá pro vývoj aplikací odlišné programovací jazyky. V následujícím přehledu nejběžnějších operačních systémů pro mobilní zařízení přiblížím jejich historický vývoj a možnosti vývoje aplikací pro jednotlivé mobilní platformy.

3.1 Android

Společnost Android Inc. byla založena v Kalifornii v říjnu 2003 Andym Rubinem, Richem Minerem, Nickem Searsem a Chrisem Whitem. V srpnu roku 2005 odkoupil tuto společnost Google Inc. a udělal z ní svou dceřinou společnost. Po odkupu tým Google Inc. vyvinul mobilní platformu založenou na Linuxovém jádře a v říjnu roku 2008 byl ve Spojených státech představen první komerční mobilní telefon s operačním systémem Android.

Android je open-source platforma, která je kompletně otevřená a umožňuje vývojářům ji volně rozšiřovat a používat ji pro různé účely. Operační systém je navržen tak, aby fungoval na širokém spektru fyzických zařízení. Díky těmto vlastnostem je Android velice oblíbená a rozšířená mobilní platforma. Vývoj tohoto operačního systému stále pokračuje a z mobilních telefonů se přesunul také do tabletů, chytrých televizorů a dalších zařízení.

Společnost Google Inc. podporuje vývoj aplikací pro tento operační systém formou rozsáhlé dokumentace, vývojových nástrojů, návodů a v neposlední řadě pomocí distribuční sítě v podobě obchodu s aplikacemi a mobilním obsahem nazvaným Google Play. Vývojáři mohou své aplikace do tohoto obchodu umisťovat a aplikace nepodléhají schvalovacímu procesu jako je tomu například u Apple App Store. Jedinou podmínkou pro vložení aplikace je registrace vývojáře jejíž součástí je registrační poplatek 500Kč.

Vyvíjet aplikace je možné v oficiálně podporovaném vývojovém prostředí Eclipse od verze 3.6.2. s pluginem Java Development Tools (JDT) a Android Development Tools (ADT). Vývoj probíhá v programovacím jazyce Java.

3.2 iOS

iOS je mobilní operační systém vyvinut společností Apple Inc. Tento operační systém byl představen v roce 2007 jako operační systém určen především pro mobilní telefon iPhone a iPod Touch. Později byl přizpůsoben pro další produkty jako iPad či Apple TV. iOS vychází z operačního systému MAC OS X a řadí se mezi Unixové operační systémy. Operační systém iOS je zcela uzavřen a je dostupný pouze pro produkty společnosti Apple Inc.

Společnost Apple Inc. podporuje vývoj aplikací formou rozsáhlé dokumentace, textových a video návodů a v neposlední řadě pomocí distribuční sítě nazvané App Store.

Tato distribuční platforma slouží nejen k distribuci aplikací, ale také veškerého digitálního obsahu jako např. knih, videí a hudby pro produkty společnosti Apple Inc., která se snaží ve všech směrech zachovat vysoké standardy a proto každá aplikace, kniha či hudba podléhá pečlivému schvalování před tím než je publikována na App Store.

Vyvíjet aplikace pro iOS je možné ve vývojovém prostředí XCode které nabízí společnost Apple Inc. a je dostupné pouze pro operační systém Mac OS X. Pro kompilaci aplikace je tedy třeba použít zařízení s operačním systémem iOS jako je např. MAC Book. Vývoj probíhá v programovacím jazyce Objective-C, který je rozšířením programovacího jazyka C.

3.3 Windows Phone

Windows Phone je operační systém vyvinutý společností Microsoft jakožto nástupce operačního systému Windows Mobile. Windows Phone byl vydán 21. října 2010 v Evropě. Tento operační systém navazuje na velice úspěšné produkty společnosti Microsoft a obsahuje jejich plnou integraci. Díky tomuto systému mobilní zařízení velice dobře spolupracují s desktopovými a cloudovými řešeními jako například Skydrive či Exchange. Microsoft podobně jako Apple se snaží sjednocovat uživatelská rozhraní na jednotlivých zařízeních a proto Windows Phone 8 se velice podobá operačnímu systému Windows 8 pro osobní počítače.

Společnost Microsoft podporuje vývoj aplikací formou rozsáhlé dokumentace, návodů, a v neposlední řadě pomocí distribuční sítě v podobě obchodu s aplikacemi a mobilním obsahem nazvaným Windows Phone Store.

Vyvíjet aplikace pro platformu Windows Phone je možné ve vývojovém nástroji Microsoft Visual Studio za pomoci SDK. Vyvíjet je možné v Microsoft .NET Frameworku a to především v Silverlightu, XNA, nebo C#.

3.4 BlackBerry

BlackBerry je mobilní operační systém vyvinutý společností Research In Motion. Tento operační systém byl vyvinut pro použití ve stejnojmenných chytrých telefonech. BlackBerry není tak rozšířený jako Android či iOS, ale stal se oblíbený především u manažeru díky podpoře pokročilé firemní korespondence.

BlackBerry nemá tak rozsáhlou vývojářskou podporu jako například Android, iOS či Windows Phone ovšem jsou zde také k dispozici návody a dokumentace z které mohou vývojáři čerpat informace. BlackBerry samozřejmě poskytuje také distribuční kanál pro aplikace a digitální obsah nazvaný BlackBerry World.

Vyvíjet aplikace pro BlackBerry je možné použitím oficiálního SDK v programovacím jazyce C, nebo C++. Další zajímavé a oficiálně dostupné možnosti vývoje jsou vytváření aplikací za pomoci HTML5 a JavaScriptu, použití Adobe AIR, nebo převedením již existujících Android aplikací.

3.5 Bada

Bada byla vyvinuta a uvedena v roce 2010 společností Samsung jako platforma pro jejich chytré telefony a potencionální konkurence mobilního operačního systému Android od společnosti Google Inc. Mobilním zařízením, které tuto platformu uvedlo na trh byl chytrý telefon Samsung GT S8500 (Wave). Bada zatím nezaznamenala výraznou podporu od vývojářů aplikací, ale je plně připravena pro vývoj.

Společnost Samsung podporuje vývoj aplikací pro mobilní operační systém Bada rozsáhlou dokumentací, video návody a v poslední době pořádáním soutěží pro vývojáře. Vývojáři jsou také podporováni distribučním kanálem pro jejich aplikace nazvaným Samsung Apps.

Vyvíjet aplikace pro operační systém Bada je možné ve vývojovém nástroji, který poskytuje společnost Samsung s implementovaným SDK pro operační systém Bada. Pro vývoj aplikací jsou k dispozici programovací jazyky C, nebo C++.

3.6 Symbian

Symbian byl původně vyvíjen společností Symbian Ltd. jako systém EPOC, který se používal pro kapesní počítače Psion. Vývoj převzala Symbian Foundation v roce 2009. Symbian byl nejvíce populární mobilní operační systém až do konce roku 2010, kdy jej překonal operační systém Android. Nyní se tento operační systém již nevyvíjí ovšem řada zařízení jej stále obsahuje a bude podporován do konce roku 2016.

Vývoj aplikací pro OS Symbian je možný v programovacím jazyce C++.

3.7 WebOS

WebOS je operační systém vyvíjený jako nástupce operačního systému PalmOS společnosti Palm. V roce 2010 proběhla akvizice a vývoj převzala společnost Hewlett Packard. Tato mobilní platforma není masivně rozšířena a o jejím komerčním potenciálu se stále diskutuje.

Společnost Hewlett Packard podporuje vývoj aplikací pro mobilní operační systém WebOS rozsáhlou dokumentací a návody. Mezi oficiální programovací jazyky pro vývoj WebOS aplikací patří Java, C, C++ či Objective-C.

4 Možnosti univerzálního vývoje mobilních aplikací

Jak můžeme vidět vývoj jedné mobilní aplikace pro nejběžnější platformy může být velice časově i finančně náročný, ale především je potřeba znát každý operační systém a jím používané technologie a nástroje. Jelikož se klade čím dál větší důraz na rychlé a levné uvedení produktu na trh v co nejkompexnější podobě začaly se zkoumat možnosti jak vyvíjet mobilní aplikace s ohledem na distribuci pro různé operační systémy. Postupně tak začaly vznikat postupy a technologie, by vývoj usnadnily.

4.1 Webové aplikace

Problém jak vyvíjet aplikace pro různá mobilní zařízení trápil vývojáře již od první možnosti rozšiřovat mobilní telefony o aplikace třetích stran. Jako jedna z prvních variant multiplatformního vývoje se ukázala možnost naprogramovat aplikaci jako webovou aplikaci umístěnou na internetu se zohledněním uživatelského rozhraní pro použití v mobilních telefonech.

Tento způsob vývoje byl velice jednoduchý, ale přinášel s sebou řadu nedostatků. Prvním a nejzávažnějším byla dostupnost aplikace. Aplikace byla umístěná na vzdáleném serveru a proto vyžadovala ke spuštění a práci s ní připojení k internetu. Jelikož pokrytí, stabilita a cena mobilního internetu byla do nedávna velkou překážkou nebyly takto vytvořené aplikace dostupné pro většinu uživatelů mobilních zařízení.

Dalším závažným nedostatkem je, že takto vytvořená aplikace není nativní. Nativní aplikace jsou ty aplikace, které jsou vytvořeny za použití oficiálních vývojových nástrojů a postupů, které publikuje autor operačního systému. Aplikace, které nejsou nativní autor operačního systému nepodporuje a nebudou být umístěny do obchodu s digitálním obsahem pro danou platformu.

Jako nedostatek můžeme také vnímat, že webová aplikace není integrována do operačního systému a tedy nemá přístup k systémové API. Nemůžeme tak pracovat s kontakty, posílat zprávy, evidovat hovory, pořizovat fotografie a podobně. Z počátku také bylo nutné si zapamatovat webovou adresu pro přístup k aplikaci a tuto adresu vkládat do webového prohlížeče. Tento nedostatek byl však odstraněn vytvořením zástupce webového prohlížeče s přednastavenou webovou stránkou do hlavní nabídky chytrých telefonů.

4.2 Multiplatformní frameworky a nástroje

Multiplatformní frameworky je označení pro nástroje, které umožňují vývojářům vyvíjet a distribuovat aplikace pro různé platformy současně a za použití stejných zdrojových kódů. V praxi tak vývojář naprogramuje aplikaci v jednom programovacím jazyce, který podporuje jím zvolený framework či nástroj a pro použití v jiné platformě překompiluje aplikaci pro tuto platformu. Díky této vlastnosti jsou multiplatformní frameworky schopny ulehčit a maximálně zefektivnit vývojářům práci a minimalizovat tak náklady spojené s vývojem. Mezi známe a často používané frameworky patří Corona SDK, Titanium, Moscrif, Icenium a PhoneGap.

4.2.1 Corona SDK

Corona SDK byla vytvořena Walterem Luhem zakladatelem Corona Labs Inc. Corona SDK umožňuje vytvářet mobilní aplikace pro iPhone, iPad, a Android zařízení. Mobilní aplikace se vyvíjejí ve scriptovacím jazyce Lua, který můžeme přirovnat například k JavaScriptu. Aplikace se pak jednoduše sestaví pro dané zařízení a operační systém. Corona SDK patří mezi licencované produkty a je potřeba toto SDK zakoupit.

4.2.2 Titanium

Appcelerator Titanium je platforma pro vývoj mobilních aplikací za pomoci webových technologií. K vývoji se především používá JavaScript rozšířený o Titanium API. Titanium není pouze SDK, ale kompletní platforma kde vývoj probíhá v IDE založeném na Eclipse pojmenovaném Titanium Studio. Toto studio obsahuje vše potřebné pro vývoj, testování a kompilaci aplikací.

4.2.3 Moscrif

Moscrif je platforma pro vývoj mobilních aplikací za pomoci JavaScriptu. Nejedná se pouze o SDK ale o kompletní nástroj s vlastním vývojovým prostředím. Po instalaci spustíte vývojové prostředí a můžete ihned začít vytvářet multiplatformní aplikace. Po dokončení vývoje stačí pouze aplikaci sestavit pro požadované operační systémy a je připravena pro použití.

4.2.4 Icenium

Icenium je další z multiplatformních vývojových nástrojů pro mobilní aplikace. K vývoji se používají webové technologie jako HTML5, CSS či JavaScript. Icenium nabízí desktopové vývojové prostředí, ale také webové vývojové prostředí. Díky tomu můžeme mobilní aplikace vyvíjet všude kde máme přístup k internetu a webový prohlížeč. Icenium patří mezi licencované produkty a je potřeba zakoupit licenci.

4.2.5 PhoneGap

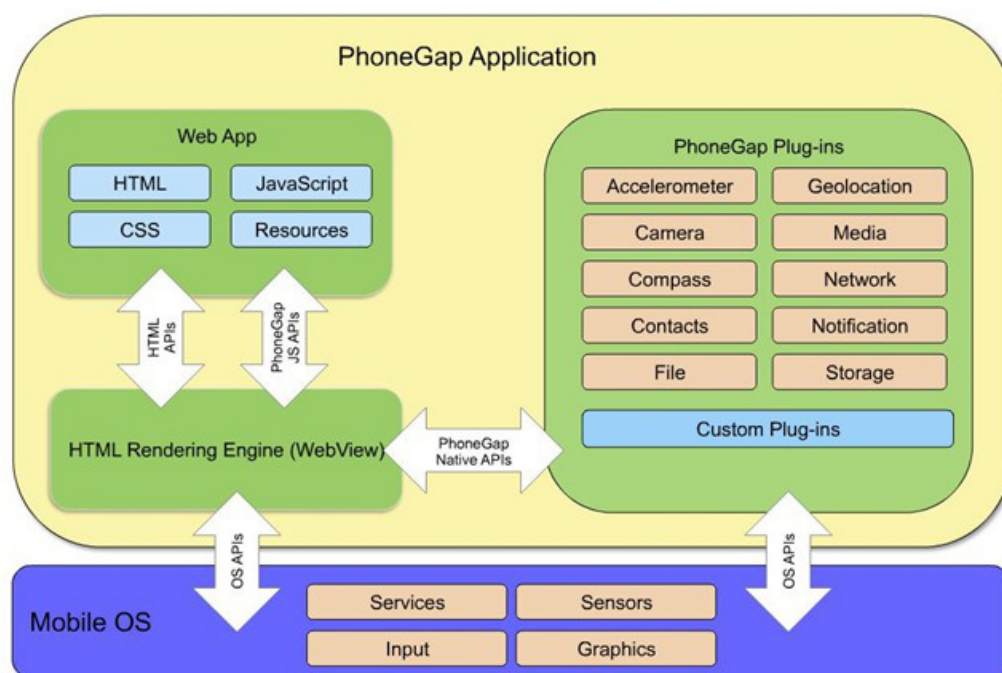
PhoneGap je framework pro vývoj multiplatformních mobilních aplikací. Jeho počátky sahají až do roku 2008 kde na konferenci iPhoneDevCamp v Sant Franciscu bylo vyvinuto jeho jádro. O rok později vyhrál cenu People's Choice Award na O'Reilly Media's 2009 Web 2.0 Conference. Po tomto úspěchu bylo v tomto frameworku vyvinuto spousta aplikací a společnost Apple Inc. začala aplikace oficiálně podporovat v App Store. V říjnu roku 2011 společnost Adobe Systems oficiálně oznámila odkup frameworku od původního vývojáře Nitobi Software a přejmenovala projekt na Apache Cordova. Obchodní název PhoneGap byl ovšem zachován.

5 Framework PhoneGap

Framework PhoneGap se řadí mezi takzvané wrappery. Wrapper funguje na principu zapouzdření zdrojového napsaného v HTML5, CSS3 a JavaScriptu spolu s vlastními knihovnami do nativního aplikačního pouzdra. Takto vytvořené aplikace se nazývají hybridní.

Framework PhoneGap tvoří komunikační rozhraní, které umožňuje mobilní aplikaci komunikovat s hardwarem a operačním systémem přístroje. Pokud vyšleme pomocí JavaScriptu požadavek na API PhoneGapu je tento požadavek poslán do integrovaného webového prohlížeče, který požadavek převede na nativní požadavek operačního systému přístroje a zašle na API operačního systému. Jakmile přístroj na požadavek odpoví odpověď opět putuje přes integrovaný webový prohlížeč a je převedena do JavaScriptu.

PhoneGap Architecture



Obrázek 2: Architektura frameworku PhoneGap [9]

Struktura Frameworku PhoneGap je rozdělena na několik pluginů podle interních funkcí mobilních zařízení např. accelerometer, kamera, kompas atd. Jejich kompletní přehled a popis funkčnosti naleznete v tabulce 1. Jelikož se každý mobilní operační systém liší nejsou tyto pluginy podporovány ve všech operačních systémech. Přehled podporovaných pluginů naleznete v tabulce 2

Název modulu	Popis
Accelerometer	Umožňuje získat informace o změně polohy přístroje vůči jeho osám
Camera	Umožňuje získat snímek z fotoaparátu
Capture	Umožňuje zaznamenávat média prostřednictvím aplikací zařízení
Compass	Umožňuje získat směr kde zařízení míří
Connection	Umožňuje přistupovat k síti internet
Contacts	Umožňuje pracovat s databází kontaktů
Device	Umožňuje získat specifické informace o zařízení
File	Umožňuje přistupovat k souborovému systému
Geolocation	Umožňuje získávat pozici pomocí GPS
Media	Umožňuje nahrávat a přehrávat zvukové soubory
Notification	Umožňuje používat vizuální, zvukové a vibrační notifikace
Storage	Umožňuje přistupovat k úložišti dat

Tabulka 1: Přehled PhoneGap API [10]

	iPhone iPhone 3G	iPhone 3GS a novější	Android	Blackberry OS 5.x	Blackberry OS 6.0x	WebOS	Windows Phone 7 + 8	Symbian	Bada
Accelerometer	ano	ano	ano	ano	ano	ano	ano	ano	ano
Camera	ano	ano	ano	ano	ano	ano	ano	ano	ano
Compass	ne	ano	ano	ne	ne	ano	ano	ne	ano
Contacts	ano	ano	ano	ano	ano	ne	ano	ano	ano
File	ano	ano	ano	ano	ano	me	ano	ne	ne
Geolocation	ano	ano	ano	ano	ano	ano	ano	ano	ano
Media	ano	ano	ano	ne	ne	ne	ano	ne	ne
Network	ano	ano	ano	ano	ano	ano	ano	ano	ano
Notification (Alert)	ano	ano	ano	ano	ano	ano	ano	ano	ano
Notification (Sound)	ano	ano	ano	ano	ano	ano	ano	ano	ano
Notification (Vibration)	ano	ano	ano	ano	ano	ano	ano	ano	ano
Storage	ano	ano	ano	ano	ano	ano	ano	ano	ne

Tabulka 2: Přehled podporovaných funkcí v operačních systémech [10]

6 Vývoj multiplatformní aplikace ve frameworku PhoneGap

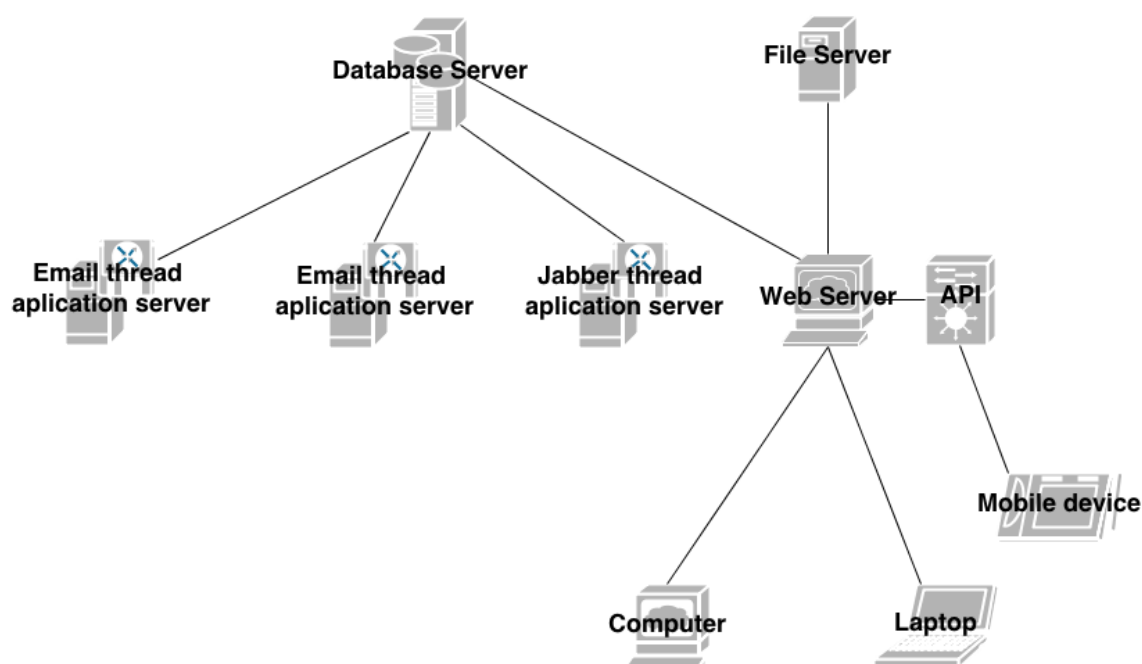
Pro vývoj ukázkové mobilní aplikace jsem se rozhodl využít webovou aplikaci, kterou vyvíjíme v rámci firemního projektu. Tato webová aplikace se skládá z několika služeb, které slouží pro řízení pokročilé komunikace s klienty. Funkcionalitou by se dala přirovnat k systémům CRM. Na tomto praktickém příkladu ukážu jak je možné poměrně rychle a s nízkými náklady vytvořit multiplatformní mobilní aplikaci již k existující webové aplikaci a rozšířit tak dostupnost aplikace o mobilní platformy.

Před zahájením vývoje jsem se seznámil s dostupnými možnostmi a nástroji jenž zle použít pro zefektivnění vývoje. Jako základní technologii pro vývoj mobilní aplikace jsem zvolil PhoneGap, který patří mezi nejdéle vyvíjené a udržované frameworky. Jedná se o velice oblíbený framework, který je vydáván jako open-source a je podporován s širokou vývojářskou komunitou. K vyvíjení mobilních aplikací se v tomto frameworku lze požit jazyky HTML5, CSS a Javascript se kterými mám již nějaké zkušenosti a také webová aplikace je v těchto jazycích vyvíjena. Vzhledem k těmto okolnostem zle také přejímat části zdrojových kódů z webové aplikace a nemusí se opětovně implementovat již vyřešené funkcionality. Framework PhoneGap se tedy jeví jako vhodný pro vytváření mobilních aplikací napojených na webové služby.

6.1 Popis webové aplikace

Webová aplikace se kterou mobilní aplikace komunikuje je rozdělena na několik základních částí. První z nich se stará o sběr kontaktů a jejich třídění do seznamů. Umožňuje tak vytvářet seznamy kontaktů, do těchto seznamů vkládat manuálně kontakty, importovat kontakty, nebo automaticky sbírat kontakty z formulářů umístěných na webu. Další část aplikace zprostředkovává emailovou komunikaci. Uživatel má možnost na vybrané kontakty rozeslat zprávy, přednastavit si plánované zaslání emailových zpráv, nebo nastavit událostmi řízenou emailovou kampaň. Součástí této části je také pokročilá analytická služba vyhodnocující emailovou komunikaci a emailové kampaně. Třetí část je zaměřena online komunikaci. Jedná se o službu poskytující možnost chatovat s návštěvníky webových stránek. Tato služba poskytuje řízení komunikace mezi návštěvníkem webových stránek a majitelem či podporou webových stránek. Služba je zaměřena na zprostředkováním komunikace mezi webovým a Jabber klientem včetně archivace komunikace a napojení na analytickou službu pro měření a vyhodnocování chatové komunikace, jejíž součástí je také sledování konverzí a chování návštěvníka webových stránek.

Webová aplikace je vyvíjena jako součást firemního projektu v Nette Frameworku a jazyce PHP s využitím databáze MySQLi. API této aplikace bylo vytvořeno pro potřeby této bakalářské práce taktéž v Nette Frameworku s ohledem na předpokládanou budoucí transformaci pomocí nástroje PHP HipHop do optimalizovaného kódu v programovacím jazyce C++.



Obrázek 3: Architektura služeb webové aplikace

6.2 Komunikace mezi webovou aplikací a mobilní aplikací

Webová aplikace nebyla z počátku navržena pro komunikaci s mobilními přístroji a ani pro distribuci na tyto platformy. Vzhledem k tomuto nedostatku bylo nutné vytvořit komunikační API díky které bude moci mobilní zařízení odesílat a přijímat data z webové aplikace. Toto API muselo být navrženo tak, aby bylo možné komunikovat napříč různými doménami, zasílat autorizované požadavky a komunikace byla zabezpečena šifrováním.

6.2.1 Možnosti komunikace

Ve frameworku PhoneGap jsou aplikace vyvíjeny v HTML5, CSS a JavaScriptu. Spouštějí se v integrovaném webovém prohlížeči na doméně localhost. Toto s sebou přináší zásadní omezení co se týče komunikace. Jelikož máme k dispozici pouze JavaScript můžeme pro komunikaci s API webové aplikace standardně využít pouze AJAX.

Jedná se o technologii, kdy pomocí JavaScriptu můžeme asynchronně načítat obsah z URL adresy. Tato technologie se jeví jako vhodná pro naše potřeby ovšem zde nastává onen problém. Z bezpečnostních důvodů není možné pomocí AJAXu načítat obsah z jiné domény než ze které byl požadavek zaslán. AJAX můžeme tedy použít pokud si přejeme získávat data z jiného zdroje v rámci aplikace např. XML souboru. Zatím co framework PhoneGap spouští naši aplikaci na doméně localhost my potřebujeme získávat data např. z domény nase-sluzba.cz a toto nám AJAX bohužel neumožní.

Pro komunikaci napříč různými doménami musíme použít JSONp. Tato technologie obchází omezení, které AJAXu brání načítání obsahu docela jednoduchým způsobem.

JSONp vychází z faktu, že omezení se nevztahuje na JavaScript, který můžeme načítat ze zdroje mimo doménu na které se nachází naše mobilní aplikace. JSONp funguje na principu dynamického vložení externího souboru s JavaScriptem, který se generuje na základě parametrů jeho adresy. Tento soubor obsahuje volání JavaScriptové callback funkce deklarované na straně odesílatele požadavku ve které jsou doplněny data, které si přejeme získat ve formátu JSON.

Pro názornou ukázkou jsem provedl základní implementaci serverové části JSONp a ukázkou volání s JavaScriptové aplikace.

```
<?php
header('Content-Type: text/javascript');
$result = getCompanyName($_GET['ic']);
echo $_GET['callback'].' ('.json_encode($result).')';

// Volání: http://sluzba.cz/api/getCompanyName/?ic=111111&callback=callback
// Vraceny obsah: callback ('Nazev spolecnosti');
```

Výpis 1: Implementace JSONp v serverové části

Po implementaci serverové části se můžeme přesvědčit zda nám vše funguje správně pomocí jednoduchého testu.

```
<script type="text/javascript">
function callback( result )
{
    alert ( result );
}
</script>
<script type="text/javascript" src="http://sluzba.cz/api/getCompanyName/?ic=111111&callback=callback">
</script>
```

Výpis 2: Jednoduché ověření fungování JSONp

Pro praktické využití je potřeba implementovat knihovnu, která se postará o dynamické vkládání elementu se scriptem a zpracování přijatých dat k tomuto účelu jsem použil Javascript framework Angular JS.

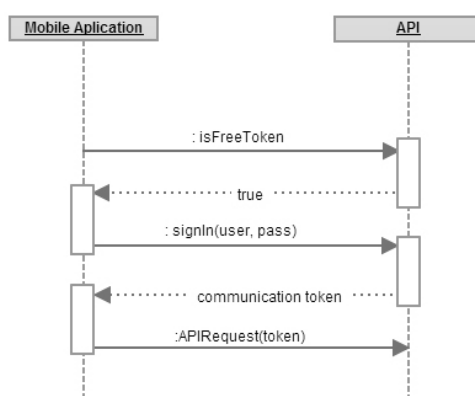
```
$scope.response = null;
$scope.url = 'http://sluzba.cz/api/getCompanyName/?ic=111111';
$http({method: 'JSONP', url: $scope.url, cache: $templateCache}).
    success(function(data, status) {
        if (data.status == '200'){
            alert (data);
        }
    }).
    error(function(data, status) {
        alert (' Connection problem');
    });
};
```

Výpis 3: Použití frameworku Angular JS pro JSONp

6.2.2 Autentizace a autorizace

Nyní když je vyřešena komunikace po technické stránce je třeba se zaměřit na autentizaci a autorizaci požadavků a v komunikaci toto zohlednit. Webová aplikace vyžaduje, aby každý uživatel, který s ní chce pracovat byl přihlášen a každý jeho požadavek byl autorizován zda je oprávněný.

Pro autentizaci v našem případě použijeme zjednodušený model (viz obr. 4) podobný OAuth. V prvním kroku klient odešle požadavek na server zda je možné se připojit. Server vrátí informaci zda ano či ne. Klient odešle uživatelské jméno, heslo a identifikační číslo mobilního zařízení. Server provede ověření zda uživatel existuje a uživatelské jméno a heslo je správné. Pokud nastane chyba při autentizaci odpoví chybovým kódem přístup zamítnut. Pokud autentizace proběhla v pořádku na základě identifikačního čísla mobilního zařízení vygeneruje komunikační token a uloží jej jako klíč ke kterému přiřadí identitu uživatele a odešle token na klienta. Klient si token uloží a pro každý požadavek, který má být autorizován odešle spolu s požadavkem také komunikační token.



Obrázek 4: Sekvenční diagram autentizace

6.2.3 Zabezpečení komunikace

Jak jsme mohli vidět tak komunikace prostřednictvím JSONp není nijak zabezpečená. Potencionální útočník může tedy jednoduše komunikaci odposlouchávat a po přečtení komunikačního tokenu sám zasílat autorizované požadavky.

Prvním krokem ke zvýšení zabezpečení by mělo být využití šifrovaného přenosu. V našem případě se jedná o rozšíření http protokolu o šifrování za využití SSL, nebo TLS což řeší https protokol. Tento protokol využívá asymetrické šifrování díky čemuž jsou posílána data nečitelná. K implementaci tohoto šifrování je potřeba zakoupit a nastavit certifikát ověřený certifikační autoritou. Https protokol řeší šifrování pouze v rámci přenosu dat a chrání nás před útokem třetí strany, která nemá přímý přístup k aplikaci.

Další možností jak zvýšit bezpečnost aplikace je šifrování posílaných dat tak, aby ani uživatelé aplikace nemohli rozpoznat názvy volaných metod a jejich parametrů. V tomto

případě je potřeba do klientské i serverové části implementovat šifrovací a dešifrovací algoritmus, který před odesláním data zašifruje a při přijetí dešifruje.

Takové jednoduché řešení může například vypadat tak, že využijeme zakódování pomocí algoritmu do formátu MIME base64. Tímto získáme běžně nečitelný řetězec, který je ovšem snadné převést zpět do čitelné podoby. Abychom tomu zabránili výsledný řetězec pomocí algoritmu aplikující bitové posuny a záměny znaků a řetězců pozměníme. Takto změněný řetězec při dekodování z MIME base64 se nezobrazí korektně. Pro dešifrování bude potřeba provést opačný postup, tedy aplikovat bitové posuny a změny znaků v opačném pořadí a až poté dekodovat z MIME base64. Útočníkovi tak zkomplikujeme čtení dat, neboť musí znát způsob jakým jsme předávaná data upravili. Bez znalosti tohoto postupu není schopen převést data do čitelné podoby.

```
http://sluzba.cz/api/getCompanyName/?ic=111111
http://sluzba.cz/api/?a=4s64sunkfxsgfs26as4fsgfsxa4xfa
```

Výpis 4: Nešifrovaný a šifrovaný požadavek

6.3 Popis mobilní aplikace

Mobilní aplikace vyvíjená v rámci bakalářské práce je zaměřena jednak na seznámení se s možnostmi propojení mobilní aplikace s již stávající webovou aplikací, ale také pro demonstrování jaké možnosti framework PhoneGap nabízí. V rámci první části byly implementovány funkce pro navázání komunikace, autentizaci, vytváření nových projektů, vytváření seznamu kontaktů, vkládání kontaktů do těchto seznamů, možnost zavolat na kontakt či poslat emailovou zprávu, zobrazit analytická data ve grafech a v nepolední řadě vytvořit klienta pro chatovou aplikaci díky kterému bude možno komunikovat s návštěvníkem webových stránek. V druhé části jsem se zaměřil na několik testů, které by měly poukázat na to, zda je možné ve frameworku PhoneGap vytvářet hry za použití HTML5 Canvasu.

6.4 Použité technologie a frameworky

Framwork PhoneGap velice dobře řeší komunikaci mezi HTML stránkami a API mobilního zařízení pomocí JavaScriptu ovšem co nepodporuje je vytváření uživatelského rozhraní aplikací a jejich aplikační logiky. Z tohoto důvodu jsem se rozhodl použít některé technologie a frameworky typické pro vývoj JavaScriptových webových aplikací.

Vzhled mobilní aplikace jsem se rozhodl vyřešit pomocí front-end frameworku Twitter Bootstrap. Pro logiku klientské aplikace jsem použil JavaScript framework Angular JS, který do aplikace vnáší MVC architekturu pro jednoduchou práci s daty, formuláři a dalšími aktivními prvky. Angular JS také obstarává JSONp komunikaci se serverovou částí. Jelikož jsem se rozhodl využít několika jQuery modulů bylo nezbytné tuto JavaScriptovou knihovnu také zahrnout do projektu. Mezi použité jQuery moduly patří Flot, který slouží pro vykreslování grafů.

6.4.1 Twitter Bootstrap

Twitter Bootstrap je front-end framework, který obsahuje řadu nástrojů, které usnadňují a urychlují vývoj webových aplikací. Jedná se především o HTML a CSS grafické prvky jako nadpisy, tabulky, tlačítka, formuláře atd., ale také o volitelné JavaScriptové doplňky. Tento front-end framework byl vyvinut ve společnosti Twitter jako nástroj pro sjednocení vzhledu jejich interních aplikací.

Twitter Bootstrap byl v bakalářském projektu použit pro vytvoření jednoduchého, přehledného a rychle implementovatelného uživatelského rozhraní.

6.4.2 Angular JS

Angular JS je JavaScriptový framework, který si klade za cíl vnést MVC architekturu do vytváření webových aplikací. Jedná se především o sadu funkcí sloužících k získávání, třídění, zpravování a následné zobrazení dat. Tento framework vyvinutý společností Google Inc. slouží především pro zjednodušení vývoje a testování aplikací založených na JavaScriptu.

Angular JS byl v bakalářském projektu použit pro získávání dat z API webové aplikace, zpracování a zobrazení těchto dat a také ke zpracování formulářů.

6.4.3 jQuery

jQuery je JavaScriptová knihovna navržena pro zjednodušení používání JavaScriptu ve webových aplikacích. Jedná se o sadu nástrojů, které jsou často a opakovaně používány při vytváření webových projektů jako například práce s DOM elementy, rozbalovací nabídky, posuvníky, obrázkové prezentace, hover nabídky atd. Tato knihovna obsahuje také řadu doplňků vytvořených vývojářskou komunitou jako je například knihovna Flot, kterou jsem použil v mobilní aplikaci.

6.4.4 Flot

jQuery knihovna Flot slouží především pro vykreslování grafů. Na základě vstupních dat je schopna vykreslit několik plně přizpůsobitelných druhů grafů. Flot jsem použil pro vykreslení grafu zobrazujícího data o doručení, přečtení a počtu stažení přílohy emailu.

6.4.5 iScroll

iScroll je JavaScriptová knihovna, která přidává možnost scrollovat u přetékajících elementů pro zařízení iPhone, iPod, iPad a operační systém Android.

6.5 Ukázková mobilní aplikace

V rámci bakalářské práce byla vyvinuta jednoduchá mobilní aplikace napojená na API webové aplikace. V této části své práce popíši jednotlivé části a jejich řešení. Uživatelské rozhraní aplikace je k nahlédnutí v příloze A.

6.5.1 Přihlášení do aplikace

Přihlášení probíhá na základě principu autentizace popsaném v kapitole 6.2.2. Uživatel zadává uživatelské jméno a heslo a odesílá požadavek na server pro ověření. Pokud vše proběhne v pořádku mobilní aplikace obdrží komunikační token, který identifikuje daného uživatele. Tento komunikační token následovně uložíme do paměti telefonu. Díky přiložení komunikačního tokenu k dalším požadavkům na API může uživatel provádět autorizované požadavky.

```
<!DOCTYPE HTML>
<html ng-app="angularjs-starter">
  <head>
    <title>Aplikace</title>
    <meta charset='utf-8'>
    <script charset="utf-8" src="js/cordova-2.5.0.js"></script>
    <script charset="utf-8" src="js/angular.min.js"></script>
    <script charset="utf-8" src="js/jquery.js"></script>
    <script charset="utf-8" src="js/bootstrap.min.js"></script>
    <script charset="utf-8" src="js/scripts.js"></script>

    <link type="text/css" rel="stylesheet" href="css/bootstrap.min.css">
    <link type="text/css" rel="stylesheet" href="css/bootstrap-responsive.min.css">
    <link type="text/css" rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <div class="container">

      <form class="form-signin" ng-controller="signIn">
        <h2 class="form-signin-heading">Please sign in</h2>
        <input type="text" ng-model="email" class="input-block-level" placeholder="Username"
          >
        <input type="password" ng-model="password" class="input-block-level" placeholder="
          Password">
        <button ng-click="signIn()" class="btn btn-large btn-primary" type="submit">Sign in</
          button>
      </form>
    </div>
  </body>
</html>
```

Výpis 5: Ukázka implementace uživatelského rozhraní

```
function signIn($scope, $http, $templateCache) {
  $scope.method = 'JSONP';

  $scope.signIn = function() {
    window.localStorage.setItem("communicationToken", "");
    $scope.code = null;
    $scope.response = null;

    $scope.url = 'http://newapi.beta-dev.cz/v2/isFreeToken?callback=JSON_CALLBACK';
    $http({method: $scope.method, url: $scope.url, cache: $templateCache}).
```

```

success(function(data, status) {
  if (data.status == '200'){
    $scope.url = 'http://newapi.beta-dev.cz/v2/signIn?login='+$scope.email+'&password'+$scope.password+'&callback=JSON_CALLBACK';
    $http({method: $scope.method, url: $scope.url, cache: $templateCache}).
    success(function(data, status) {
      if (data.status == '200'){
        // Save communication token to mobile local storage
        window.localStorage.setItem("communicationToken", data.response.token);
        window.location = "projects.html";
      } else if (data.status == '401'){
        alert (' Authentication failed ');
      } else {
        alert (' Internal Server Error ');
      }
    }).
    error(function(data, status) {
      alert (' Connection problem');
    });
  } else if (data.status == '401'){
    alert (' No free token ');
  } else {
    alert (' Internal Server Error ');
  }
}).
error(function(data, status) {
  alert (' Connection problem');
});

});
}

```

Výpis 6: Ukázka implementace aplikační logiky

6.5.2 Správa projektů

Projekty slouží pro rozdělení uživatelského účtů na další podskupiny. Každý projekt může symbolizovat jinou firmu, webovou stránku, nebo jak název napovídá projekt. Mobilní aplikace umožňuje vytvářet nové projekty a s jednotlivými projekty dále pracovat. Seznam projektů získáme zavoláním metody API `getProjects` s jediným parametrem jímž je komunikační token. Přijatá data ve formátu JSON pak pomocí Angular JS jednoduše zobrazíme. K vytvoření nového projektu je potřeba zavolat metodu API `createProject` s parametry název projektu, popis projektu, email a komunikačním tokenem.

6.5.3 Správa kontaktů

Správa kontaktů umožňuje vkládat nové kontakty a stávajícím kontaktům zatelefonovat či odeslat email. Funkce telefonování doplní telefonní číslo s aplikace do mobilního

přístroje. Funkce zaslání emailové zprávy odešle zprávu prostřednictvím API do emailové fronty, která je následně obsluhována emailovými servery webové aplikace. Emaily nemohou být zasílány přímo z mobilního zařízení neboť je potřeba obsah emailu upravit a doplnit o analytické prvky a také zajistit úspěšné doručení. Emailová fronta a servery jenž ji obsluhují jsou navrženy tak, aby zvládali rozesílat velké množství emailu v poměrně krátkém čase a také zabezpečit, aby emaily nebyly označovány jako spam ¹.

6.5.4 Statistiky emailů

Další částí aplikace je přehled odeslaných emailů a statistiky jejich zobrazení a stažení příloh. API poskytuje vstupní data pro generování grafů pomocí JavaScript knihovny Flot. Jelikož jsou přílohy ukládány na CDN ² a v emailu presentovány pomocí odkazu je možné měřit jejich stažení na rozdíl od klasických příloh.

6.5.5 Zákaznický chat

Zákaznický chat umožňuje komunikovat se zákazníky na webových stránkách prostřednictvím chatovací platformy. Klientská aplikace na pozadí periodicky přistupuje k webové aplikace API a získává data o nové konverzaci a tyto data zobrazuje. Mobilní aplikace umožňuje také odesílat zprávy na chatovou aplikaci umístěnou na webových stránkách, která taktéž periodicky kontroluje nové zprávy a zobrazuje je.

6.5.6 Canvas generátor částic a hra Snake

Jako zatěžkávací test frameworku PhoneGap a hlavně JavaScriptu jsem zvolil vygenerování fontánky z 2000 částic na Canvas. Tato poměrně náročná animace na výpočty ve webovém prohlížeči běžela hladce, ovšem v mobilním zařízení se viditelně zasekávala. Framework PhoneGap tedy není vhodný na programování složitějších grafických aplikací a her používajících převážně Canvas pro vykreslování animací.

Dalším testem bylo použití klasické hry Snake naprogramované v HTML5 a JavaScriptu. Migrace této hry do frameworku PhoneGap proběhl bez problémů a velice snadno. Jelikož PhoneGap podporuje HTML5 i Javascript stačilo pouze překopírovat zdrojové kódy a napojit ovládací prvky. Co se týče výkonu hra běžela hladce a bez potíží.

Hledal jsem cestu jak změřit výkon všech 3 aplikací a porovnat je. Jako jediná možnost změření výkonu aplikací se jevílo porovnat hodnotu snímků za sekundu. Maximální hodnota tohoto ukazatele je stanovena na 60 snímků za sekundu. Ve webovém prohlížeči Google Chrome na osobním počítači se 4 jádrovým procesorem a 4GB RAM aplikace pro správu kontaktů a hra Snake vykazovaly maximální hodnotu 60 snímků za sekundu. Generátor 2000 částic vykázal hodnotu, která se přibližovala 30 snímkům za sekundu. Na mobilním zařízení Samsung Galaxy Ace aplikace pro správu kontaktů a hra Snake

¹SPAM je označení pro nevyžádanou poštu.

²Služba zajišťující distribuci a synchronizaci dat na serverech na různých kontinentech. Takto synchronizovaná data jsou kdekoliv k dispozici s velmi krátkou odezvou

vykazovaly podobné hodnoty mezi 55 - 60 snímky za sekundu. Generátor částic již nedopadl tak dobře a vykazoval hodnoty okolo 2,2 snímků za sekundu.

Název aplikace	FPS v Google Chrome	FPS v Samsung Galaxy Ace
Správa kontaktů	60	58
Hra Snake	60	55
Generátor částic	30	2,2

Tabulka 3: Přehled výkonu jednotlivých aplikací

Na základě těchto třech testovacích aplikací můžeme usoudit, že omezením nejsou vlastnosti JavaScriptu, ale náročnost aplikací, výkon použitého frameworku a výkon zařízení na kterém jsou spouštěny.

Zatím co u osobního počítače vykreslování pomáhá výkonná grafická karta u mobilního zařízení viditelně tento výkon postrádáme. Jelikož však zpomalení vykreslování částic není ve stejném poměru jako je tomu u osobního počítače pravděpodobně bude hrát velkou roli rychlost komunikace mezi mobilním telefonem a aplikací prostřednictvím frameworku PhoneGap.

7 Kompilace mobilní aplikace pro různé operační systémy

7.1 Kompilace aplikace pro operační systém Android

Jak už bylo zmíněno framework PhoneGap umožňuje multiplatformní vývoj mobilních aplikací, ale aby bylo možné aplikaci zkompileovat je nutné nainstalovat nativní vývojové prostředí a nastavit projekt, do kterého budou umístěny multiplatformní zdrojové soubory.

7.1.1 Instalace potřebných vývojových nástrojů a knihoven

Jako první věc, kterou potřebujeme pro sestavení projektu je vývojový nástroj Eclipse Classic, Android SDK, ADT Plugin a framework PhoneGap pro Android. Pokud všechny tyto nástroje a knihovny jsou nainstalovány můžeme pokračovat vytvořením projektu.

7.1.2 Vytvoření projektu

Projekt vytvoříme běžným způsobem. Po spuštění aplikace Eclipse z hlavní nabídky vybereme File - New - Android project. Po zobrazení průvodce pokračujeme volbou názvu, podporované verze Androidu a nastavením názvu balíčku. Po dokončení průvodce je vytvořen běžný projekt Android aplikace.

7.1.3 Konfigurace projektu pro používání frameworku PhoneGap

Pro úspěšné začlenění frameworku PhoneGap do projektu je třeba provést některé úpravy. První z nich je vytvoření správné adresářové struktury. Vytvoříme dvě složku /assets/www a složku /libs. Nyní dekomprimujeme archív s frameworkem PhoneGap a přejdeme do složky lib/android v této složce nalezneme veškeré potřebné soubory. Zkopírujeme soubor cordova-1.5.0.js do složky /assets/www v našem projektu. Dále pak soubor cordova-1.5.0.jar do složky /libs v našem projektu a nakonec složku xml do složky /res v našem projektu.

Pokud již máme mobilní aplikaci vytvořenou pouze ji zkopírujeme do složky /assets/www. Pokud však aplikaci ještě nemáme vytvoříme testovací soubor index.html ve složce /assets/www do kterého vložíme kostru PhoneGap aplikace.

```
<html>
  <head>
    <title>PhoneGap</title>
    <script type="text/javascript" charset="utf-8" src="cordova-1.5.0.js"></script>
  </head>
  <body>
    <h1>Hello PhoneGap</h1>
  </body>
</html>
```

Výpis 7: Kostra PhoneGap aplikace

Dále je potřeba přidat knihovnu cordova-1.5.0.jar do projektu tak, aby byla součástí sestavené aplikace. Toto provedeme kliknutím pravým tlačítkem myši na knihovnu a z nabídky vybereme Build Path - Add to Build path.

Nyní je potřeba upravit třídu hlavní aktivity tak, aby spouštěla PhoneGap. Po úpravě bude třída hlavní aktivity vypadat přibližně následovně.

```
package com.example.bakalarka;

import android.os.Bundle;
import org.apache.cordova.DroidGap;

public class MainActivity extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

Výpis 8: Třída hlavní aktivity Android aplikace

Na závěr je ještě potřeba upravit nastavení aplikace a doplnit potřebná oprávnění. Tyto nastavení se provádí v souboru AndroidManifest.xml, který by měl mít po změnách přibližně tuto podobu.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bakalarka"
    android:versionCode="1"
    android:versionName="1.0" >

    <supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:resizeable="true"
        android:anyDensity="true"
    />

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="16" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:configChanges="orientation|keyboardHidden"
            android:name="com.example.bakalarka.MainActivity"
            android:label="@string/app_name" >
```

```

<intent-filter>
  <action android:name="android.intent.action.MAIN" />

  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>

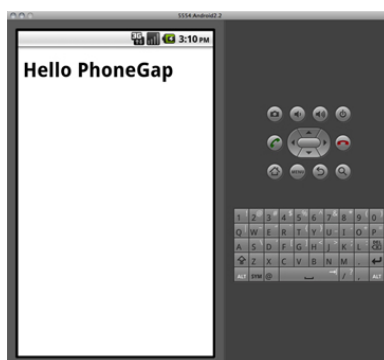
<activity
  android:name="org.apache.cordova.DroidGap"
  android:label="@string/app_name"
  android:configChanges="orientation|keyboardHidden">
  <intent-filter></intent-filter>
</activity>
</application>

<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"
  "/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
</manifest>

```

Výpis 9: AndroidManifest.xml

Nyní stačí aplikaci pouze sestavit a spustit. V emulátoru, který Eclipse nabízí bude aplikace vypadat nějak takto (viz obr. 5).



Obrázek 5: Hello World! PhoneGap aplikace

7.2 Kompilace pro operační systém Windows Phone

Aby bylo možné sestavit aplikaci pro Windows Phone bude potřeba nainstalovat vývojové prostředí Visual Studio a Windows Phone SDK.

7.2.1 Sestavení šablony pro Visual Studio

Abychom měli ve Visual Studiu k dispozici šablonu ze které budeme vytvářet PhoneGap projekty musíme ji nejdříve sestavit. Je potřeba otevřít soubor `/lib/windows-phone/templates/standalone/CordovaSolution.sln`, který nalezneme v archívu s frameworkem PhoneGap ve Visual Studiu Express pro Windows Phone. Z menu vybereme File - Export Template. Z nabídky zvolíme Project Template. V další nabídce je potřeba nechat zatrženou volbu Automatically import the Template a dokončit vytvoření šablony tlačítkem Finish.

7.2.2 Vytvoření nového PhoneGap projektu

Nyní když máme sestavenou šablonu můžeme jednoduše vytvořit mobilní aplikaci využívající framework Phonegap jako projekt. Z hlavní nabídky File vybereme New - Project a zvolíme CordovaWP8AppStandalone. Nyní máme vytvořený PhoneGap Hello World! projekt. Námi vytvořenou aplikaci pak stačí zkopírovat do složky www a sestavit.

7.3 Problémy a jejich řešení

Při vývoji prostřednictvím frameworku PhoneGap se objevily některé problémy a nedostatky, které bylo nutné vyřešit.

7.3.1 Cross-domain komunikace pomocí JavaScriptu

Hlavním problémem bylo vyřešit cross-domain komunikaci pomocí JavaScriptu. Vzhledem k bezpečnostním omezením nelze cross-domain komunikovat pomocí AJAXu a bylo nutné se ohlédnout po jiné technologii. V rámci bakalářského projektu jsem využil JSONp jehož princip byl zde již objasněn v kapitole 6.2.1.

7.3.2 Nepodporované scrollování pomocí CSS vlastnosti overflow: auto;

Mobilní zařízení mají velké problémy s přetékáním blokových elementů a jejich scrollování pomocí css vlastnosti overflow nastavenou na hodnotu auto či scroll. Většina mobilních zařízení tuto vlastnost nepodporuje a není možné v elementech scrollovat. V rámci mé mobilní aplikace jsem tento problém řešil pomocí JavaScript knihovny iScroll, která doplní scrollování postavené na JavaScriptu. Použití vypadá pak takto.

```
myScroll = new iScroll('idelementu');
```

Výpis 10: Použití knihovny iScroll

8 Závěr

Při zpracovávání bakalářské práce jsem se seznámil v několika vývojovými nástroji a frameworky pro multiplatformní vývoj mobilních aplikací. Na základě mých zkoumání jednotlivých nástrojů konstatuji, že každý nástroj je vhodný pro jiný typ projektu. V této práci jsem ukázkovou mobilní aplikaci vytvářel nad API již existující webové aplikace. Pro tento typ projektu jsem zvolil framework PhoneGap ve kterém je možné přejímat zdrojové kódy z webové aplikace díky jeho podpoře HTML5, CSS a JavaScriptu. Díky této vlastnosti nebylo například nutné vytvářet nové formuláře v HTML a JavaScript scripty, ale daly se použít formuláře a scripty z již existující aplikace což značně urychlilo vývoj.

V případě vývoje her, či aplikací orientovaných na práci s mobilním přístrojem bych nejspíše použil Corona SDK, které je sice komerční ale obsahuje podporu 3D grafiky a dle prezentace tvůrce je zaměřeno poměrně rozsáhle na vývoj her.

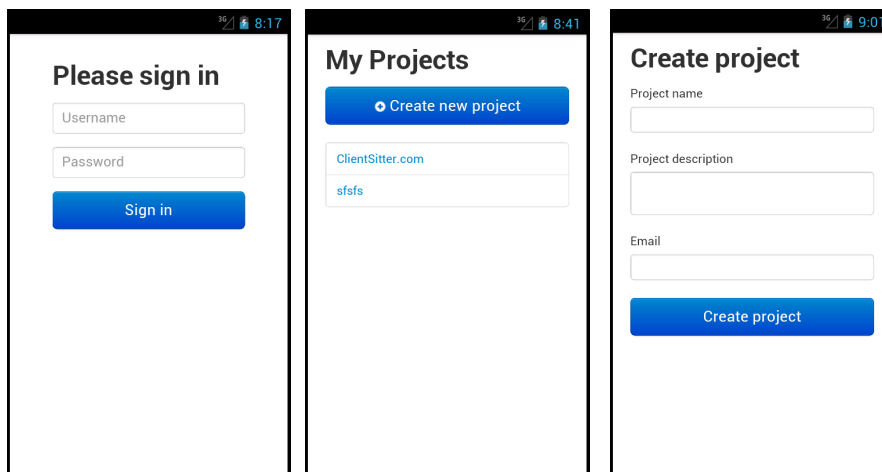
Vývoj mobilních aplikací pomocí nástroje pro multiplatformní vývoj se jeví jako logický krok pro zefektivnění vývoje. Multiplatformní nástroje mohou pomoci s optimalizací jak časových tak finančních nákladů, které jsou u vývoje pro každou jednotlivou platformu viditelně vyšší. Pokud se podíváme na hry vyvíjené v těchto nástrojích můžeme si všimnout, že již pronikly mezi vývojářské společnosti a některé z nich patří k velice oblíbeným a velmi často instalovaným.

Multiplatformní vývoj mobilních aplikací vidím jako cestu jak rychle, efektivně a levněji vytvořit aplikaci pro širokou škálu různých mobilních zařízení a operačních systémů.

9 Reference

- [1] WARGO, John M., *PhoneGap Essentials: Building Cross-Platform Mobile Apps*, NY: Addison-Wesley Pub. Co., 2012.
- [2] LUNNY, Andrew, *Phonegap beginner's guide: build cross-platform mobile applications with the PhoneGap open source development framework*, UK: Packt Publishing Limited, 2011.
- [3] PILGRIM, Mark, *Dive into HTML5*, <http://www.diveintohtml5.info/>
- [4] FLING, Brian, *Mobile design and development 1st ed.*, CA: O'Railly, 2009
- [5] NEIL, Theresa, *Mobile design pattern gallery 1st ed.*, CA: O'Railly, 2012
- [6] BRUNETTE, E. Hello, *Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers)*, Pragmatic Bookshelf, 2010
- [7] Titanium SDK & Titanium Studio, In a.appcelerator <http://docs.appcelerator.com/titanium/latest/>
- [8] Ecommerce Study Finds Mobile Safari To Be Fastest Growing Web Browser <http://techcrunch.com/2012/06/01/ecommerce-study-finds-mobile-safari-to-be-fastest-growing-web-browser/>
- [9] PhoneGap: avagy a mobil fejlesztés leghatékonyabb HTML5/JavaScript-ben? <http://html5hu.wordpress.com/2011/10/05/a-mobil-fejleszt-es-leghatkonyabb-html5javascript-ben/>
- [10] PhoneGap.com - Supported Features <http://phonegap.com/about/feature/>
- [11] DALLERA, Andrea, Why you should stay away from Appcelerator's Titanium, In Andrea Dallera 's blog <http://usingimho.wordpress.com/2011/06/14/why-you-should-stay-away-from-appcelerators-titanium/>
- [12] FREDERICK, G. R., *Beginning Smartphone Web Development: Building JavaScript, CSS, HTML and Ajax-based Applications for iPhone, Android, Palm Pre, BlackBerry, Windows Mobile and Nokia S60*. Apress, Apress, 2010
- [13] HUME, Tom App.ft.com, and the cost of cross-platform web apps <http://www.tomhume.org/2011/10/appftcom-and-the-cost-ofcross-platform-web-apps.html>
- [14] PhoneGap Beliefs, Goals, and Philosophy <http://phonegap.com/2012/05/09/phonegap-beliefs-goals-and-philosophy/>
- [15] Corona SDK <http://www.coronalabs.com/products/corona-sdk/>
- [16] Moscrif <http://moscrif.com/>
- [17] Icenium <http://www.icenium.com/>

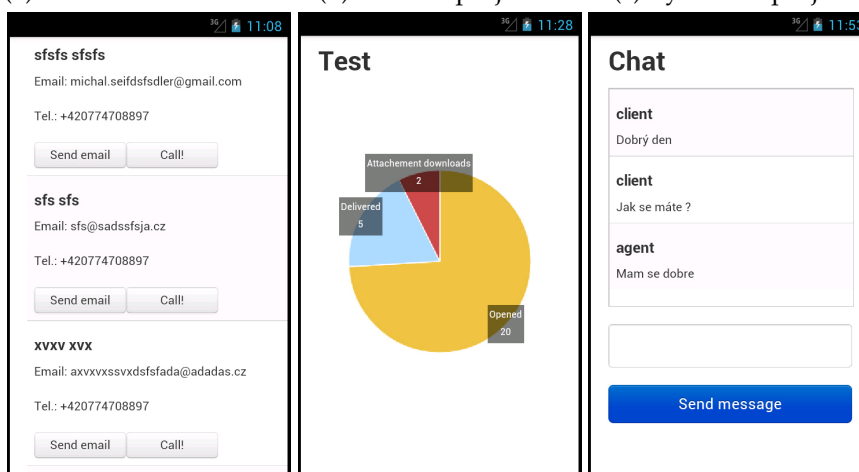
A Uživatelské rozhraní aplikace



(a) Přihlášení uživatele

(b) Seznam projektů

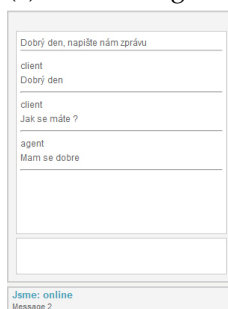
(c) Vytvoření projektu



(d) Výpis kontaktů

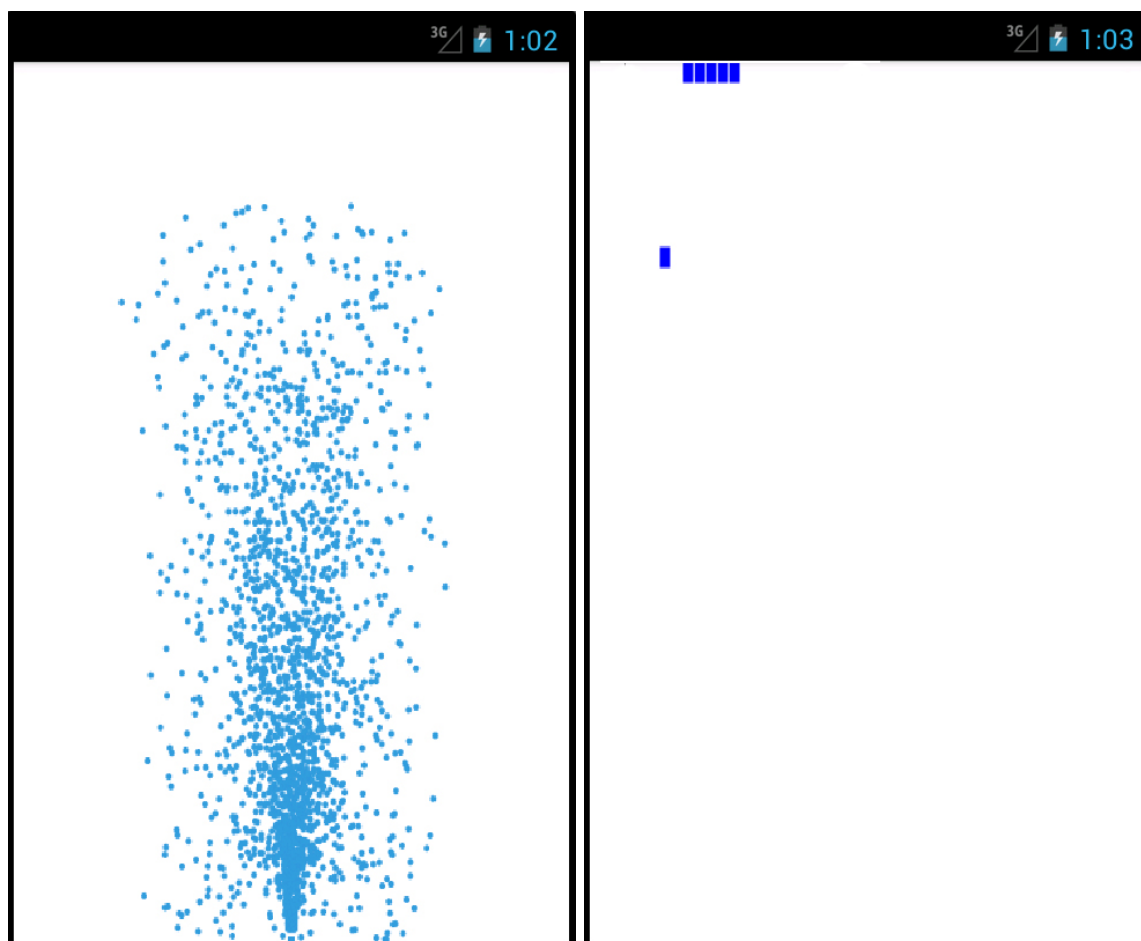
(e) Zobrazení grafu

(f) Chat



(g) Chat na webu

Obrázek 6: Uživatelské rozhraní



(a) Canvas 2000 částic

(b) Canvas hra Snake

Obrázek 7: Ukázka kreslení na Canvas

B Příloha na CD/DVD

- Elektronická podoba bakalářské práce
- Zdrojové kódy ukázkové mobilní aplikace
- Prezentace k bakalářské práci